IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	)	
	)	Atty Docket No: SUNMP068
PABLA	)	
	)	
Application No: 09/201,644	)	Examiner: SAX, S.
	)	
Filed: November 30, 1998	)	Group Art Unit: 2174
	)	
For: METHOD AND APPARATUS FOR	)	
DETECTING DEVICE SUPPORT IN A	)	Date: July 30, 2004
GRAPHICAL USER INTERFACE	)	

TRANSMITTAL OF APPEAL BRIEF
(PATENT APPLICATION -- 37 CFR 192)

Commissioner for Patents
Mail Stop: Appeal Brief- Patents
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on April 27, 2004. The Notice of Appeal was received by the USPTO on April 29, 2004. Therefore, the due date for this Appeal Brief is August 29, 2004, with a two-month extension of time. This Appeal Brief is transmitted in triplicate:

This application is on behalf of:

☐ Small Entity ☒ Large Entity

Pursuant to 37 CFR 1.17(f), the fee for filing the Appeal Brief is:

☐ $165.00 (Small Entity) ☒ $330.00 (Large Entity)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136

apply:

☒ Applicant petitions for an extension of time under 37 CFR 1.136 (fees: 37 CFR 1.17(a)-(d)) for the total number of months checked below:

Months	Large Entity	Small Entity
☐ one	$110.00	$55.00
☒ two	$420.00	$210.00
☐ three	$950.00	$475.00

☒ If an additional extension of time is required, please consider this a petition therefor.

☐ An extension for __ months has already been secured and the fee paid therefor of $ is deducted from the total fee due for the total months of extension now requested.

☐ Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that Applicant has inadvertently overlooked the need for a petition and fee for extension of time.

Total Fees Due:

Appeal Brief Filing Fee	$330.00
Extension Fee (if any)	$420.00
Total Fee Due	**$750.00**

☒ Enclosed is Check No. 12169 in the amount of $750.00.
☒ Charge any additional fees or credit any overpayment to Deposit Account No. 50-0850, (Order No. SUNMP068). Two copies of this transmittal are enclosed.

Respectfully submitted,
MARTINE & PENILLA, LLP

Kenneth D. Wright
Registration No. 53,795

710 Lakeway Drive, Suite 170
Sunnyvale, CA 94085
(408) 749-6900
Customer No. 32,291

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE PABLA

Application for Patent

Filed November 30, 1998

Application No. 09/201,644

FOR:
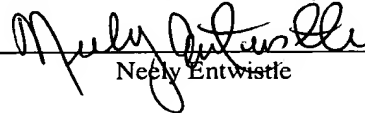
METHOD AND APPARATUS FOR DETECTING DEVICE SUPPORT IN A GRAPHICAL USER INTERFACE

APPEAL BRIEF

08/03/2004 EAREGAY1 00000041 09201644

01 FC:1402   330.00 OP

MARTINE & PENILLA, LLP
Attorneys for Applicant

TABLE OF CONTENTS

APPENDIX A - CLAIMS ON APPEAL

I. REAL PARTY IN INTEREST

The real party in interest is Sun Microsystems, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The undersigned is not aware of any related appeals and/or interferences.

III. STATUS OF THE CLAIMS

A total of 37 claims were presented during prosecution of this application. Claims 2, 5-6, 9-10, and 12-21 were cancelled during prosecution of this application. The Applicant appeals rejected claims 1, 3-4, 7-8, 11, and 22-37.

IV. STATUS OF THE AMENDMENTS

The application was originally filed on November 30, 1998. A continued prosecution application (CPA) of the original application was filed on September 11, 2000. A request for continued examination (RCE) of the CPA was filed August 6, 2002. All amendments have been entered, leaving rejected claims 1, 3-4, 7-8, 11, and 22-37.

V. SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for detecting device support in a graphical user interface (GUI). (p. 16, 1st ¶) More specifically, embodiments of the present invention include techniques for detecting support for a given input device by a screen element of a GUI. (p. 16, 2nd ¶) In one embodiment, a runtime version of a screen element's program code is examined to detect an ability to process an input device's events. (p. 16, 2nd ¶) In another embodiment, an operation is performed at runtime to

determine whether a screen element of a GUI has delegated processing of a given input device's events to other program code. (p. 16, 2nd ¶) In yet another embodiment, a runtime version of a screen element's program code is examined to detect a declaration of program code that is indicative of a screen element's support or non-support of a given input device. (p. 16, 2nd ¶) Furthermore, it should be appreciated that one or more of the above mentioned embodiments can be combined. (p. 16, 2nd ¶)

In accordance with the present invention, an appearance of a screen element in a GUI can be modified according to the determination as to whether the screen element provides support for a particular input device. (p. 16, last ¶) Further, an identification of support for an input device by the screen element can facilitate an identification of the screen element, where it is otherwise difficult to determine which screen element is to receive an input device's event. (p. 16, last ¶) For example, where a pointing device (e.g., mouse) is positioned within a region of the GUI that is occupied by more than one screen element, an event of the pointing device can be sent to the screen element that has the ability to process the pointing device's event. (p. 17, 1st partial ¶)

In accordance with at least one embodiment of the invention, screen elements are object-oriented objects written using the Java programming language. (p. 19, 1st ¶) Runtime versions of screen elements can be examined to determine whether methods for handling a given type of input device are present. (p. 19, 1st ¶) In the Java programming language, the runtime version of the screen element is a bytecode version of a class definition of the screen element. (p. 19, 2nd ¶) Therefore, when considering a screen element in the Java programming language, the corresponding bytecode class definition is examined to determine whether the definition includes at least one "device-handling" method. (p. 19, 2nd ¶) Also, an object class may inherit methods from another class (e.g., a superclass). (p. 19, 3rd ¶) Therefore, in a case where the screen element's class definition

does not include an input device method, embodiments of the invention further examine the screen element's superclass bytecode object class definition to determine whether the superclass definition includes at least one "device-handling" method. (p. 19, 3rd ¶)

For example, to detect whether a screen element supports mouse input, method inspection is performed on the screen element's bytecode class definition to detect at least one "mouse-handling" method. (p. 20, 1st full ¶) If at least one mouse-handling method is detected in the screen element's bytecode class definition, the screen element is marked as being able to support mouse input. (p. 20, 2nd full ¶) An indication that the screen element can support mouse input may also be used to modify the look and feel of the screen element in the GUI. (p. 20, 2nd full ¶)

Figure AB-1 is an illustration showing an example of a method inspection process flow, in accordance with one embodiment of the present invention. (p. 20, last ¶) The process flow examines a class definition for a method that processes a device's input, e.g., such as a mouse event method. (p. 20, last ¶) If a device method exists in the class definition, the screen element is assumed to support the device. (p. 20, last ¶) In one embodiment of the invention, the process flow may be executed for each of a number of device methods until one of the device methods is found. (p. 21, 1st partial ¶)

At step 302, a screen element's class definition becomes a current class definition. (p. 21, 1st full ¶) At step 304, the current class definition is examined to determine whether a given input device method exists in the definition. (p. 21, 1st full ¶) If so, processing continues at step 312 to mark the screen element as supporting input for the given input device. (p. 21, 1st full ¶) Following step 312, processing ends at step 316. (p. 21, 1st full ¶)

If the given input device method is not found at step 304, processing continues at step 308 to get a superclass of the current class. (p. 21, 2nd full ¶) The superclass becomes the current class. (p. 21, 2nd full ¶) At step 310, a determination is performed to determine

if the current class is the originating superclass, e.g., java.awt.Component. (p. 21, 2nd full ¶) If so, processing continues at step 314 to mark the screen element as not supporting input for the given input device. (p. 21, 2nd full ¶) Following step 314, processing ends at step 316. (p. 21, 2nd full ¶) If it is determined at step 310 that the current class is not the originating superclass, processing returns to step 304 to examine the current class definition for the given input device method. (p. 21, 2nd full ¶)

In one embodiment, a source object includes methods to register "listeners" with the source object. (p. 24, 2nd full ¶) In this embodiment, processing that would otherwise be directed to a source object is directed to a listener object that has registered with the source object. (p. 24, 2nd full ¶) For example, the listener object may be activated by the source object to handle input device events. (p. 24, 2nd full ¶) A source object can register itself and/or another object as a listener. (p. 24, 2nd full ¶) Thus, a source object may activate itself where it is a registered listener. (p. 24, 2nd full ¶) In this embodiment, the present invention provides for marking a screen element as supporting a given input device when the screen element has a listener that supports the given input device. (p. 24, 3rd full ¶)

In another embodiment of the present invention, the input device method detection process is performed when a screen element's object instance is constructed, i.e., in a constructor method. (p. 22, 1st full ¶) In this embodiment, the input device method detection process may be performed in a constructor method of a platform-independent object, a peer object, or a platform-dependent object. (p. 22, 1st full ¶)

```
                    ┌─────────┐
                    │  START  │
                    └─────────┘
                         │
                         ▼
              ┌──────────────────────┐  ⌐ 302
              │     SET SCREEN       │
              │   ELEMENT'S CLASS    │
              │   AS CURRENT CLASS   │
              └──────────────────────┘
                         │
                         ▼
                     ◇ 304                         ┌────────────────────┐  ⌐ 312
        ┌───┐      ╱ METHOD  ╲        YES          │   SCREEN ELEMENT   │
        │ 1 │────▶│ EXISTS IN CURRENT │──────────▶ │  SUPPORTS DEVICE   │
        └───┘      ╲   CLASS?   ╱                   └────────────────────┘
                     ◇                                        │
                         │                                    │
                         │ NO                                 │
                         ▼          ⌐ 308                      │
              ┌──────────────────────┐                        │
              │  SET CURRENT CLASS   │                        │
              │  TO BE SUPERCLASS    │                        │
              │  OF CURRENT CLASS    │                        │
              └──────────────────────┘                        │
                         │                                    │
                         ▼                                    │
                     ◇ 310                         ┌────────────────────┐  ⌐ 314
              ╱ CURRENT CLASS ╲       YES          │   SCREEN ELEMENT   │
             │ IS ORIGINATING   │───────────────▶ │  DOES NOT SUPPORT  │
              ╲  SUPERCLASS? ╱                     │      DEVICE        │
                     ◇                              └────────────────────┘
                         │                                    │
                         │ NO                                 ▼
                         ▼                            ┌─────────┐  ⌐ 316
                       ┌───┐                          │   END   │
                       │ 1 │                          └─────────┘
                       └───┘
```
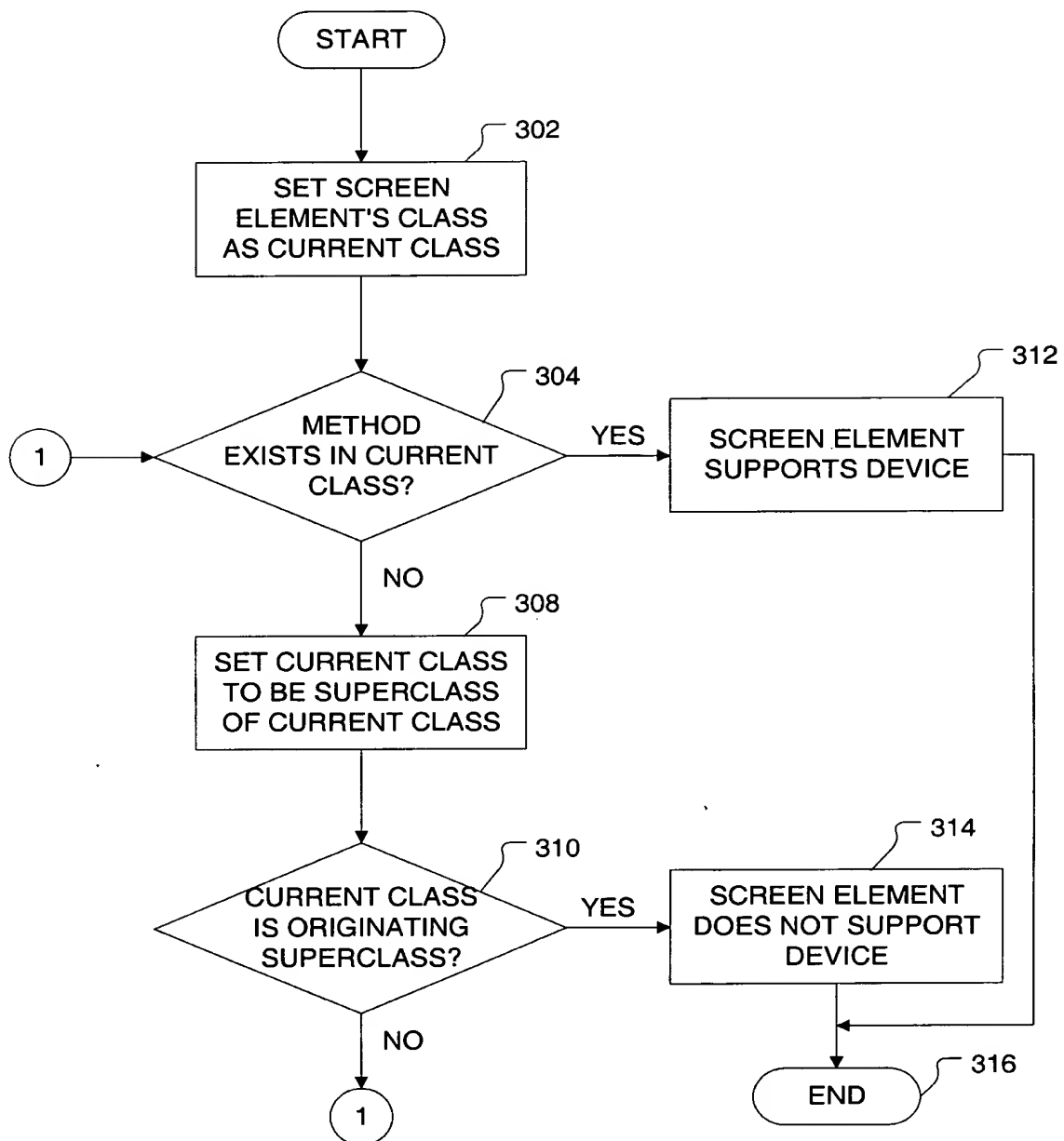
Fig. AB-1

VI. ISSUES

The issues presented in this appeal are whether the rejections under 35 U.S.C. §103 of the claims under appeal are proper. The issues therefore are as follows:

A.   Are claims 1, 3-4, 7-8, 11, and 22-37 properly rejected under 35 U.S.C. §103(a) as being unpatentable over Finch et al. ("Finch") (U.S. Patent No.: 5,805,796) and Ashe et al. ("Ashe") (U.S. Patent No.: 6,307,574) and Guillen et al. ("Guillen") (U.S. Patent No.: 5,701,485)?

VII. GROUPING OF THE CLAIMS

The applicant proposes three groups of claims. The first group (Group I) includes claims 1, 3-4, 7-8, 23, and 26-31. The claims of the first group stand or fall together. The second group (Group II) includes claims 11, 24-25, and 32-33. The claims of the second group stand or fall together. The third group (Group III) includes claims 22 and 34-37. The claims of the third group stand or fall together.

VIII. ARGUMENTS

A.   The references as relied upon by the Examiner, either separately or in combination, do not motivate or suggest to one of ordinary skill in the art at the time of the invention to combine the reference teachings in a manner that would render the invention as recited in claims 1, 3-4, 7-8, 23, and 26-31 (Group I) prima facie obvious.

Rejection

Claims 1, 3-4, 7-8, 23, and 26-31 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

With regard to claim 1, the Examiner's position is as follows:

"Regarding claim 1, Ashe et al. show examining the program code of a screen element of a GUI (column 3 lines 10-20, column 6 lines 10-25) wherein

examining is performed upon execution of the GUI (column 5 lines 7-24), and without execution of the class definition (column 5 lines 5-14), and identifying an element if the program code includes a method supporting the element (column 6 lines 5-10 and 34-55). Ashe et al. do not specifically state the element is supporting an input device, but does use class definitions to determine support for an element, for analysis and control of the gui system. Furthermore, Finch et al. do determine the element is supporting an input device (column 5 lines 60-68 and column 6 lines 1-20), in a system using class definitions for analysis and control of a GUI system (column 8 lines 29-45). It would have been obvious to a person with ordinary skill in the art to have Ashe et al. determine an element supporting an input device, because it would provide convenient analysis and control of a GUI in a system that uses class definitions for analysis and control of a GUI. In addition, Finch et al. and Ashe et al. do not go into the details of the superclass definition being examined if the element is identified as not supporting (the input device), but Ashe et al. do mention examining the class for functionality. Furthermore, Guillen et al. show examining a superclass definition if the element is not supporting a functionality (column 2 lines 18-24, 40-45; column 4 lines 38-55; column 5 lines 48-60; column 6 lines 9-19). This is done to efficiently examine a class for functionality. It would have been obvious to a person with ordinary skill in the art to have this in the system described by Ashe et al. in view of Finch et al. as explained above, because it would be an efficient way to examine a class for functionality."

Applicant's Rebuttal

Claim 1 represents the broadest independent claim of Group I (i.e., claims 1, 3-4, 7-8, 23, and 26-31). Since the claims of Group I will stand or fall together, the Applicant

chooses to argue the patentability of claim 1. Therefore, the arguments presented in this section (Section VIII.A.) will be directed to claim 1.

Ashe teaches a method by which program code relating to GUI elements can be organized into a multi-level structure of classes. A class defining a structure and a function of the GUI element resides at one level within the multi-level structure of classes. A class defining an appearance of the GUI element resides at another level within the multi-level structure of classes. Only one instance of the class defining the structure and the function of the GUI element is required, regardless of the number of instances of the class defining the appearance of the GUI element. In this manner, Ashe teaches the implementation of multiple appearance themes for a GUI element without having to repeat the structure and function class definitions for each theme.

Finch discloses a diagnostic system capable of representing physical devices as software objects derived from a generic base class. A number of device classes are derived from the generic base class. Each device class is distinguished by a characteristic definition. Also, a number of diagnostic device objects are derived from the generic base class. Each diagnostic device object is associated with a corresponding device class and a particular physical device. Additionally, each diagnostic device object has an encapsulated device characteristic definition corresponding to physical characteristics of the particular physical device.

Guillen discloses a system for dispatching messages between instance specific dispatch tables of objects, when a particular object does not have a method called for execution in response to a message received by the particular object. When a message is sent to a first object and the method called for execution is not resident or associated with the first object, the first object may expressly reference another object of the same class which may contain the required method. The message is accordingly dispatched to another

instance specific dispatch table associated with an object which may be able to execute the method called for by the initially transmitted message.

The method of Ashe for organizing program code relating to GUI elements into a multi-level structure of classes is not related to either Finch's diagnostic system or Guillen's message dispatching system. Also, there is no relation between Finch's diagnostic system and Guillen's message dispatching system. Furthermore, neither Ashe, Finch, Guillen, nor the combination thereof, teach or suggest detection of input device support of a screen element of a GUI, as recited in claim 1. With respect to claim 1, as presented in Appendix A of this Appeal Brief, the Examiner has asserted that the claimed features are present in the cited art of record. However, the Applicant has not found any teaching or suggestion of the features of claim 1 in the cited art of record.

Notwithstanding the lack of relevant teachings in either Ashe, Finch, or Guillen, the Applicant submits that there is no suggestion or motivation, either explicitly or implicitly, in either Ashe, Finch, or Guillen to have combined their respective teachings to arrive at the present invention as embodied in claim 1. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art. MPEP §2143.01 The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

The Examiner has stated that the motivation to combine the asserted teachings of Ashe, Finch, and Guillen is as follows:

"It would have been obvious to a person with ordinary skill in the art to have Ashe et al. determine an element supporting an input device, because it would provide convenient analysis and control of a GUI in a system that uses class definitions for analysis and control of a GUI."

"It would have been obvious to a person with ordinary skill in the art to have this in the system described by Ashe et al. in view of Finch et al. as explained above, because it would be an efficient way to examine a class for functionality."

Contrary to the Examiner's interpretation, claim 1 is not simply directed to examining a class for functionality. Rather, claim 1 defines a method for detecting input device support of a screen element in a GUI. Claim 1 requires examination of program code associated with the screen element to detect an ability or inability to process an input device event. The ability or inability to process the input device event is contingent on the presence of input device-handling code. If the screen element is determined to support the input device, the screen element is marked accordingly.

The Applicant submits that neither Ashe, Finch, Guillen, nor the combination thereof, provide the motivation to combine their respective teachings as stated by the Examiner. The rationale for combining references requires a recognition either expressly or impliedly in the prior art or drawn from a convincing line of reasoning based on established scientific principles or legal precedent, that some advantage or expected beneficial result would have been produced by the combination of references. *In re Sernaker*, 702 F.2d 989, 994-95, 217 USPQ 1, 5-6 (Fed. Cir. 1983). The Applicant submits that the references neither expressly or impliedly provide the motivation to combine their respective teachings as suggested by the Examiner. Furthermore, the Applicant submits that the motivation to combine the teachings of Ashe, Finch, and Guillen as suggest by the

Examiner is not drawn from a convincing line of reasoning based on established scientific principles or legal precedent.

Furthermore, the test for obviousness is what the combined teachings of the references would have suggested to one of ordinary skill in the art. MPEP §2143.01 However, the level of ordinary skill in the art cannot be relied upon to provide the suggestion to combine references. *Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 50 USPQ2d 1161 (Fed. Cir. 1999). A statement that modifications of the prior art to meet the claimed invention would have been within the ordinary skill of the art at the time the claimed invention was made is not sufficient to establish a prima facie case of obviousness without some objective reason to combine the teachings of the references. *Ex parte Levengood*, 28 USPQ2d 1300 (Bd. Pat. App. & Inter. 1993).

In view of the foregoing, the Board of Appeals and Interferences ("Board") is respectfully requested to overrule the Examiner's rejections of claims 1, 3-4, 7-8, 23, and 26-31 under 35 U.S.C. §103.

B.   The references as relied upon by the Examiner, either separately or in combination, do not motivate or suggest to one of ordinary skill in the art at the time of the invention to combine the reference teachings in a manner that would render the invention as recited in claims 11, 24-25, and 32-33 (Group II) prima facie obvious.

Rejection

Claims 11, 24-25, and 32-33 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

With regard to claim 11, the Examiner's position is as follows:

"Regarding claim 11, in addition to the aforementioned [with respect to claim 1], the system of Finch et al. may use JAVA bytecode. It would have been

obvious to a person with ordinary skill in the art to use JAVA bytecode, because it would be a convenient language in which to define a screen element."

Applicant's Rebuttal

Claim 11 represents the broadest independent claim of Group II (i.e., claims 11, 24-25, and 32-33). Since the claims of Group II will stand or fall together, the Applicant chooses to argue the patentability of Claim 11. Therefore, the arguments presented in this section (Section VIII.B.) will be directed to claim 11.

The Applicant's Rebuttal to Examiner's Position as discussed above in Section VIII.A., with respect to claim 1, is equally applicable to the present discussion of claim 11. To minimize redundancy, the Board is respectfully requested to refer to Section VIII.A. above for the present Applicant's Rebuttal to Examiner's Position.

In view of the foregoing, the Board is respectfully requested to overrule the Examiner's rejections of claims 11, 24-25, and 32-33 under 35 U.S.C. §103.

C.   The references as relied upon by the Examiner, either separately or in combination, do not motivate or suggest to one of ordinary skill in the art at the time of the invention to combine the reference teachings in a manner that would render the invention as recited in claims 22 and 34-37 (Group III) prima facie obvious.

Rejection

Claims 22 and 34-37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

With regard to claim 22, the Examiner's position is as follows:

"Claims 22-37 show the same features as above [claims 1, 3-4, 7-8, and 11] and are rejected for the same reasons."

Applicant's Rebuttal

Claim 22 represents the broadest independent claim of Group III (i.e., claims 22 and 34-37). Since the claims of Group III will stand or fall together, the Applicant chooses to argue the patentability of claim 22. Therefore, the arguments presented in this section (Section VIII.C.) will be directed to claim 22.

The Applicant's Rebuttal to Examiner's Position as discussed above in Section VIII.A., with respect to claim 1, is equally applicable to the present discussion of claim 22. To minimize redundancy, the Board is respectfully requested to refer to Section VIII.A. above for the present Applicant's Rebuttal to Examiner's Position.

In view of the foregoing, the Board is respectfully requested to overrule the Examiner's rejections of claims 22 and 34-37 under 35 U.S.C. §103.

D.   The combination of Finch, Ashe, and Guillen, as relied upon by the Examiner, fail to teach or suggest all features recited in each of claims 1, 3-4, 7-8, 23, and 26-31 (Group I) as required to establish a prima facie case of obviousness.

Rejection

Claims 1, 3-4, 7-8, 23, and 26-31 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

To minimize redundancy, the Board is respectfully requested to refer to Section VIII.A. above for the Examiner's Position.

Applicant's Rebuttal

Since the claims of Group I will stand or fall together, the Applicant chooses to argue the patentability of claim 1. Therefore, the arguments presented in this section (Section VIII.D.) will be directed to claim 1.

With regard to claim 1, the Examiner asserts that Ashe shows examining the program code of a screen element of a GUI (column 3, lines 10-20 and column 6, lines 10-25) wherein examining is performed upon execution of the GUI (column 5, lines 7-24), and identifying an element if the program code includes a method supporting the element (column 6, lines 5-10 and 34-55). Ashe (particularly column 3, lines 10-20 and column 6, lines 10-25) does not teach or suggest <u>examining program code associated with a screen element of a GUI to detect an ability to process an input device event</u>. Ashe simply discloses a multi-level class structure for separating the code associated with functionality and structure of the GUI element from the code associated with appearance of the GUI element. Thus, Ashe teaches a method by which the class definitions of a GUI element are organized. It is apparent that Ashe's teachings are not relevant to <u>examining program code associated with a screen element of a GUI to detect an ability to process an input device event</u>.

Further with respect to claim 1, the Examiner asserts that identifying an element if the program code includes a method supporting the element is taught by Ashe, column 6, lines 5-10 and 34-55. The subject portion of the claim reads as follows: "automatically identifying the screen element as supporting the input device when input device-handling program code is detected within the program code associated with the screen element, the input device-handling program code signifying the ability to process the input device event." It is not clear that the Examiner's statement of "a method supporting the element" is equivalent to the claim language of "input device-handling program code signifying the ability to process the input device event." Notwithstanding this ambiguity, Ashe (column 6, lines 5-10 and 34-55) still fails to teach or suggest this particular element of the claimed invention. Ashe (column 6, lines 5-10 and 34-55) teaches that each separate appearance of a control object has its own definition with its own associated program code. Ashe also

teaches the use of core control classes to implement the basic functionality and overall appearance of the control objects. In the context of Ashe, there is no teaching relevant to the subject feature of claim 1, e.g., Ashe does not teach examining the core control classes associated with the control object to identify the associated GUI element if the class definition includes a method supporting an input device's input. Thus, the teachings of Ashe (column 6, lines 5-10 and 34-55) are not relevant to the presently claimed invention.

Further with respect to claim 1, the Examiner states the following: "Ashe et al. do not specifically state the element is supporting an input device, but does use class definitions to determine support for an element, for analysis and control of the gui system." The applicant agrees that Ashe does not specifically or inferentially state, teach, or suggest examining program code associated with a screen element of a GUI to detect an ability to process an input device event. Contrary to the Examiner's position, there are no teachings or suggestions in Ashe related to determining support for an element. Ashe simply teaches the use of class definitions organized in a hierarchical manner to provide for efficient multiple themed implementation of GUI elements such as a control objects or menus. To determine support for an input device or to detect an ability to process an input device's events are both objectives that require specific actions to be accomplished. The mere fact that Ashe, or any other reference, uses class definitions to implement a screen element of a GUI does not imply that there is a determination or detection being made of the class's capabilities with respect to processing input device events. Such a determination or detection involves specific and focused actions that are not taught or suggested by Ashe, but are claimed by the present invention.

Additionally, the Examiner has referred to Ashe (column 5, lines 1-13) to assert that the screen element is marked if the class definition includes support for the input device. However, Ashe does not teach "marking the screen element in response to

automatically identifying the screen element as supporting the input device, wherein program code associated with the screen element has been examined to detect an ability to process an input device event. In contrast to the claimed invention, Ashe teaches changing the appearance of the screen element in response to user execution of the screen element.

Further with respect to claim 1, the Examiner asserts that Finch determines that the element is supporting an input device (column 5, lines 60-68 and column 6, lines 1-20), in a system using class definitions for analysis and control of a GUI system (column 8, lines 29-45). The Applicant submits that Finch simply does not teach or suggest any part of claim 1. Furthermore, the cited portions of Finch as relied on by the Examiner do not even appear to be pertinent to the presently claimed invention.

The Examiner has admitted that neither Finch nor Ashe go into the details of the superclass definition being examined if the element is identified as not supporting the input device. The Examiner has relied upon Guillen (column 2, lines 18-24 and 40-45, column 4, lines 38-55, column 5, lines 48-60, and column 6, lines 9-19) to teach the following feature of claim 1:

> "examining program code associated with a preceding superclass of the screen element to detect the ability to process the input device event, wherein examining the program code associated with the preceding superclass is performed in response to automatically identifying the screen element as not supporting the input device."

Guillen teaches a method by which object instances are created which inherit dispatch tables of their classes and superclasses and additionally annex instance-specific dispatch tables of their own. The teachings of Guillen as referenced by the Examiner have absolutely nothing to do with examining program code associated with a preceding superclass of a screen element for the purpose of detecting an ability to process an input

device event. Furthermore, Guillen is completely silent with respect to examining the program code associated with the superclass in response to automatically identifying the screen element as not supporting the input device. The only commonality between Guillen and the features of claim 1 is that they both use the term superclass. Therefore, Guillen does not teach the features of claim 1 as asserted by the Examiner.

Though the shortcomings of Ashe, Finch, and Guillen have been discussed above according to the Examiner's application of the cited art, the Applicant submits that the combination of Ashe, Finch, and Guillen fails to render claim 1 prima facie obvious. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). As discussed above, the combination of Ashe, Finch, and Guillen fail to teach or suggest all the features of claim 1 as necessary to establish a case of prima facie obviousness against claim 1.

Furthermore, the Applicant submits that the Examiner has attempted to attack the recited features of claim 1 in a piecemeal manner without considering the claimed invention as a whole. In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983). Therefore, the Examiner must consider all features of the claimed invention. Furthermore, "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

In view of the foregoing, the Board is respectfully requested to overrule the Examiner's rejections of claims 1, 3-4, 7-8, 23, and 26-31 under 35 U.S.C. §103.

E.   The combination of Finch, Ashe, and Guillen, as relied upon by the Examiner, fail to teach or suggest all features recited in each of claims 11, 24-25, and 32-33 (Group II) as required to establish a prima facie case of obviousness.

Rejection

Claims 11, 24-25, and 32-33 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

With regard to claim 11, the Examiner's position is as follows:

"Regarding claim 11, in addition to the aforementioned [with respect to claim 1], the system of Finch et al. may use JAVA bytecode. It would have been obvious to a person with ordinary skill in the art to use JAVA bytecode, because it would be a convenient language in which to define a screen element."

Applicant's Rebuttal

Since the claims of Group II will stand or fall together, the Applicant chooses to argue the patentability of claim 11. Therefore, the arguments presented in this section (Section VIII.E.) will be directed to claim 11.

The Examiner has essentially stated that, with exception of the Java bytecode feature, claim 11 is rejected under the same basis of rejection as applied to claim 1. Therefore, the Applicant respectfully requests the Board to refer to Section VIII.D. for the Applicant's arguments with respect to features of claim 11 that are common with features of claim 1.

With respect to the Java bytecode feature of claim 11, however, the Examiner has simply stated that "the system of Finch et al. may use JAVA bytecode." The Examiner has not clearly identified how Finch teaches of suggests "examining Java bytecode defining the screen element of the graphical user interface to detect an ability of the screen element to

process a signal to be received from an input device." The Examiner has stated that merely defining a screen element in JAVA is obvious. The Applicant submits that the Java bytecode limitation of claim 11 is not merely provided to define a screen element. Rather, claim 11 requires the Java bytecode to be examined to detect an ability of the screen element to process a signal to be received from an input device. Therefore, the method of the present invention as embodied in claim 11 infers a capability to examine Java bytecode for the purpose of detecting an ability of a screen element to process a signal to be received from an input device.

The Applicant submits that the combination of Ashe, Finch, and Guillen are silent with respect to Java bytecode as recited in claim 11. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). As discussed above and in Section VIII.D., the combination of Ashe, Finch, and Guillen fail to teach or suggest all the features of claim 11 as necessary to establish a case of prima facie obviousness against claim 11.

In view of the foregoing, the Board is respectfully requested to overrule the Examiner's rejections of claims 11, 24-25, and 32-33 under 35 U.S.C. §103.

F. The combination of Finch, Ashe, and Guillen, as relied upon by the Examiner, fail to teach or suggest all features recited in each of claims 22 and 34-37 (Group III) as required to establish a prima facie case of obviousness.

Rejection

Claims 22 and 34-37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Finch and Ashe and Guillen. These rejections are traversed.

Examiner's Position

With regard to claim 22, the Examiner's position is as follows:

"Claims 22-37 show the same features as above [claims 1, 3-4, 7-8, and 11] and are rejected for the same reasons."

Applicant's Rebuttal

Since the claims of Group III will stand or fall together, the Applicant chooses to argue the patentability of claim 22. Therefore, the arguments presented in this section (Section VIII.F.) will be directed to claim 22.

The Board is respectfully requested to refer to Sections VIII.D. and VIII.E. for the Applicant's arguments with respect to features of claims 1 and 11, respectively, that are common with features of claim 22.

Additionally, the Applicant submits that the Examiner has not indicated how the combination of Ashe, Finch, and Guillen teach "examining a set of instructions for operating a region of the graphical user interface to detect an ability to respond when input is received from an input device, the region of the graphical user interface containing one or more screen elements." Also, the Examiner has not indicated how the cited art of record teaches or suggests "determining which of the one or more screen elements is associated with the one or more input device-handling instructions."

The Examiner has not addressed the above-mentioned regional aspect of examining the set of instructions. Also, the Examiner has not addressed determining which of the one or more screen element in the region is associated with the input device-handling instructions. Therefore, the Examiner has failed to indicate how the prior art teaches or suggests all of the claim limitations. Yet again, to establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Furthermore, "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

In view of the foregoing, the Board is respectfully requested to overrule the Examiner's rejections of claims 22 and 34-37 under 35 U.S.C. §103.

G.   Conclusion

In view of the inappropriateness of the 35 U.S.C. §103 rejections, as discussed in the Applicant's aforementioned arguments, the Applicant submits that the presently claimed invention is patentable over the cited art of record.

The Applicant respectfully requests the Board to consider each group of claims (Groups I, II, and III) separately with respect to the teachings of the cited art of record.

In sum, the Applicant submits that the Examiner's rejections are in error, and respectfully requests that the Board of Appeals and Interferences reverse the Examiner's rejections of the claims on appeal.

Respectfully Submitted,
MARTINE & PENILLA, LLP

Kenneth D. Wright
Reg. No. 53,795

MARTINE & PENILLA, LLP
710 Lakeway Drive, Suite 170
Sunnyvale, California 94085
408.749.6900

APPENDIX A

CLAIMS ON APPEAL

1.   In a computer system, a method of detecting input device support of a screen element of a graphical user interface, comprising:

examining program code associated with a screen element of a graphical user interface to detect an ability to process an input device event, wherein the examining is performed upon execution of the graphical user interface;

automatically identifying the screen element as supporting the input device when input device-handling program code is detected within the program code associated with the screen element, the input device-handling program code signifying the ability to process the input device event;

marking the screen element in response to automatically identifying the screen element as supporting the input device;

automatically identifying the screen element as not supporting the input device when input device-handling program code is not detected within the program code associated with the screen element; and

examining program code associated with a preceding superclass of the screen element to detect the ability to process the input device event, wherein examining the program code associated with the preceding superclass is performed in response to automatically identifying the screen element as not supporting the input device.

3.   The method in accordance with claim 1, wherein marking the screen element in response to automatically identifying the screen element as supporting the input device includes modifying a look of the screen element.

4.   The method in accordance with claim 1, wherein examining program code associated with the screen element is performed during construction of an object instance of the screen element.

7.   The method in accordance with claim 1, wherein examining program code associated with the screen element includes examining one or more interface declarations associated with the screen element.

8.   The method in accordance with claim 7, wherein the one or more interface declarations are contained in an implements clause.

11.   In a computer system, a method of determining input device support of a screen element of a graphical user interface, comprising:

executing the graphical user interface;

examining Java bytecode defining the screen element of the graphical user interface to detect an ability of the screen element to process a signal to be received from an input device;

automatically identifying a portion of the Java bytecode defining the screen element as having the ability to process the signal to be received from the input device; and

altering an appearance of the screen element to signify the ability to process the signal to be received from the input device.

22.   In a computer system, a method of detecting input device support of a graphical user interface, comprising:

examining a set of instructions for operating a region of the graphical user interface

to detect an ability to respond when input is received from an input device, the region of

the graphical user interface containing one or more screen elements;

identifying one or more input device-handling instructions in the set of instructions;

determining which of the one or more screen elements is associated with the one or

more input device-handling instructions; and

altering an appearance of the one or more screen elements associated with the one

or more input device-handling instructions to signify the ability to respond when input is

received from the input device.

23.   The method in accordance with claim 1, wherein the input device-handling

program code is a listener object registered with the screen element.

24.   In a computer system, a method of determining input device support of a

screen element of a graphical user interface as recited in claim 11, wherein the screen

element represents a lightweight element capable of functioning independently from a

native layer.

25.   In a computer system, a method of determining input device support of a

screen element of a graphical user interface as recited in claim 11, further comprising:

automatically identifying an inability of the screen element to process the signal to

be received from the input device; and

altering the appearance of the screen element to signify the inability to process the

signal to be received from the input device.

26. A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface, comprising:

program instructions for examining program code associated with a screen element of a graphical user interface to detect an ability to process an input device event, wherein the program instructions are defined to direct the examining of the program code to be performed upon execution of the graphical user interface;

program instructions for automatically identifying the screen element as supporting the input device when input device-handling program code is detected within the program code associated with the screen element, the input device-handling program code signifying the ability to process the input device event;

program instructions for marking the screen element in response to automatically identifying the screen element as supporting the input device;

program instructions for automatically identifying the screen element as not supporting the input device when input device-handling program code is not detected within the program code associated with the screen element; and

program instructions for examining program code associated with a preceding superclass of the screen element to detect the ability to process the input device event, wherein the program instructions are defined to direct the examining of the program code associated with the preceding superclass to be performed in response to automatically identifying the screen element as not supporting the input device.

27. A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface as recited in claim 26, wherein the program instructions for marking the screen element in response to

automatically identifying the screen element as supporting the input device includes program instructions for modifying a look of the screen element.

28.   A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface as recited in claim 26, wherein the program instructions for examining program code associated with the screen element direct the examining to be performed during construction of an object instance of the screen element.

29.   A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface as recited in claim 26, wherein the program instructions for examining program code associated with the screen element includes program instructions for examining one or more interface declarations associated with the screen element.

30.   A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface as recited in claim 29, wherein the one or more interface declarations are contained in an implements clause.

31.   A computer readable media containing program instructions for detecting input device support of a screen element of a graphical user interface as recited in claim 26, wherein the input device-handling program code is a listener object registered with the screen element.

32.   A computer readable media containing program instructions for determining input device support of a screen element of a graphical user interface, comprising:

program instructions for examining Java bytecode defining the screen element of the graphical user interface to detect an ability of the screen element to process a signal to be received from an input device;

program instructions for automatically identifying a portion of the Java bytecode defining the screen element as having the ability to process the signal to be received from the input device; and

program instructions for altering an appearance of the screen element to signify the ability to process the signal to be received from the input device.

33.   A computer readable media containing program instructions for determining input device support of a screen element of a graphical user interface as recited in claim 32, further comprising:

program instructions for automatically identifying an inability of the screen element to process the signal to be received from the input device; and

program instructions for altering the appearance of the screen element to signify the inability to process the signal to be received from the input device.

34.   A computer readable media containing program instructions for detecting input device support of a graphical user interface, comprising:

program instructions for examining a set of instructions for operating a region of the graphical user interface to detect an ability to respond when input is received from an input device, the region of the graphical user interface containing one or more screen elements;

program instructions for identifying one or more input device-handling instructions in the set of instructions;

program instructions for determining which of the one or more screen elements is associated with the one or more input device-handling instructions; and

program instructions for altering an appearance of the one or more screen elements associated with the one or more input device-handling instructions to signify the ability to respond when input is received from the input device.

35.   A computer comprising:

a display for displaying at least one screen element of a graphical user interface;

at least one input device; and

a detector configured to examine program code associated with the at least one screen element to detect an ability to process an input signal to be received from the at least one input device, the detector further configured to alter a display of the at least one screen element to signify that the at least one screen element is capable of processing signals to be received from the at least one input device.

36.   A computer as recited in claim 35, wherein the detector includes program code readable by a processor of said computer.

37.   A computer as recited in claim 35, wherein the detector is further configured to examine program code associated with the at least one screen element to detect an inability to process the input signal to be received from the at least one input device, the detector further configured to alter the display of the at least one screen element

to signify that the at least one screen element is incapable of processing signals to be received from the at least one input device.